FIG. 1

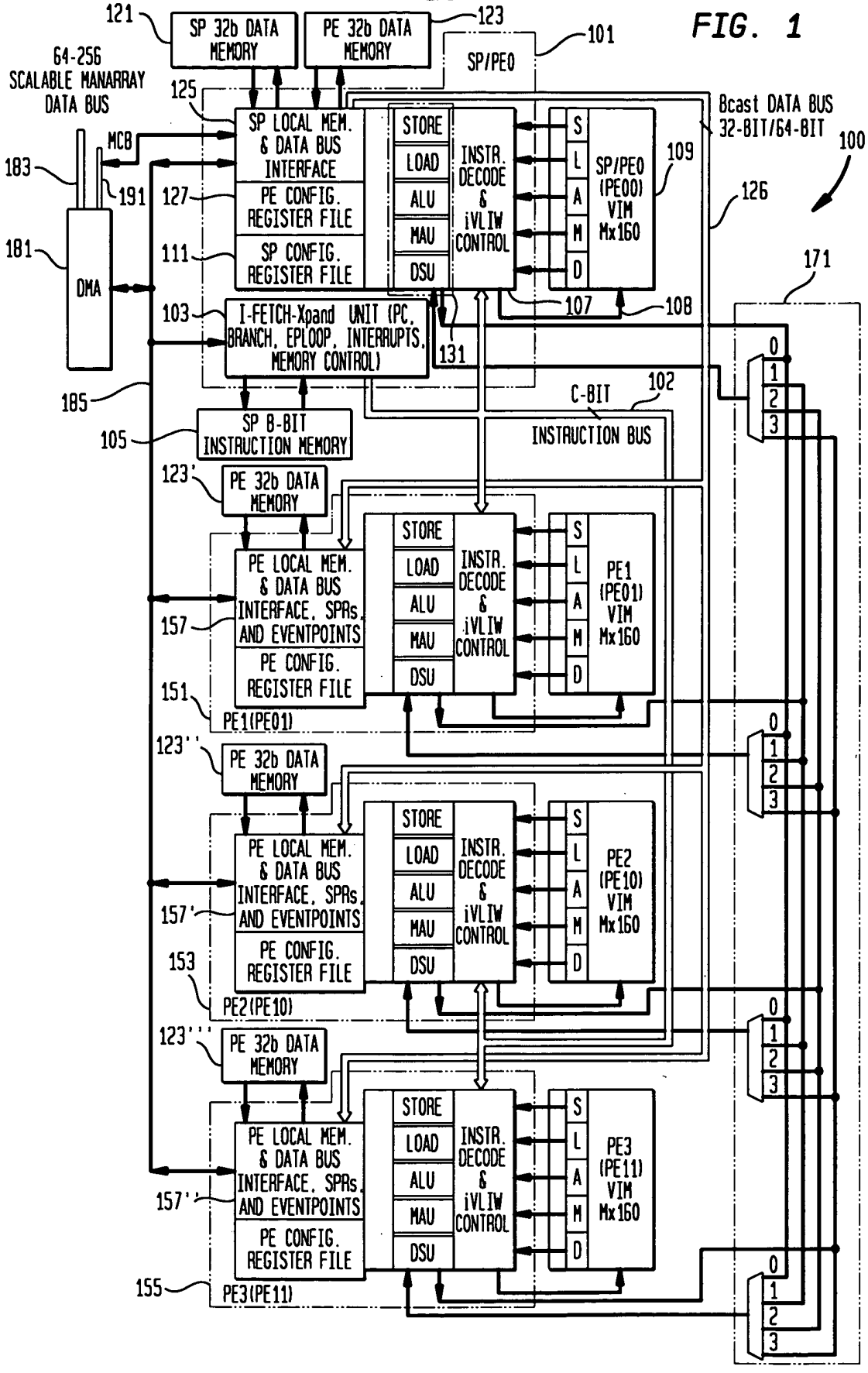
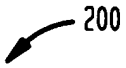
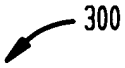

121
SP 32b DATA MEMORY
123
PE 32b DATA MEMORY
101
SP/PE0
64-256 SCALABLE MANARRAY DATA BUS
125
SP LOCAL MEM. & DATA BUS INTERFACE
STORE
LOAD
ALU
MAU
DSU
INSTR. DECODE & iVLIW CONTROL
S
L
A
M
D
SP/PE0 (PE00) VIM Mx160
109
Bcast DATA BUS 32-BIT/64-BIT
100
MCB
183
191
127
PE CONFIG. REGISTER FILE
126
181
111
SP CONFIG. REGISTER FILE
DMA
171
103
I-FETCH-Xpand UNIT (PC, BRANCH, EPLOOP, INTERRUPTS, MEMORY CONTROL)
107
108
131
185
C-BIT
102
105
SP 8-BIT INSTRUCTION MEMORY
INSTRUCTION BUS
123'
PE 32b DATA MEMORY
PE LOCAL MEM. & DATA BUS INTERFACE, SPRs, AND EVENTPOINTS
157
PE CONFIG. REGISTER FILE
PE1 (PE01) VIM Mx160
151
PE1(PE01)
123''
PE 32b DATA MEMORY
157'
PE2 (PE10) VIM Mx160
153
PE2(PE10)
123'''
PE 32b DATA MEMORY
157''
PE3 (PE11) VIM Mx160
155
PE3(PE11)
0
1
2
3

FIG. 2



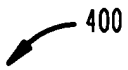

```
foreach instruction {add mpy sub} {
  # for each processor
  foreach p  {s p} {
    # for each conditional execution
    foreach ce {e t. f.} {
      # for each unit that the instruction holds
      foreach unit {a m d} {
        # foreach testvector
        foreach columns in answerset {
          # generate test with these parameters
        }
      }
    }
  }
}
```

FIG. 3




```
set instruction(MPY,FORMAT,1sw)      {RTE RX RY}
set instruction(MPY,FORMAT,1uw)      {RTE RX RY}
set instruction(MPY,FORMAT,2sh)      {RTE RX RY}
set instruction(MPY,FORMAT,2uh)      {RTE RX RY}
set instruction(MPY,RFACCESS)        {{WRITE} {READ} {READ}}
set instruction(MPY,DATATYPES)       {1sw 1uw 2sh 2uh}
set instruction(MPY,DIFFDATATYPES)   {{1sw 1sd 1sw 1sw}{1uw 1ud 1uw 1uw}\
                                     {2sh 2sw 2sh 2sh}{2uh 2uw 2uh 2uh}}
set instruction(MPY,PROCS)           {s p}
set instruction(MPY,UNITS)           {m}
set instruction(MPY,CE)              {e t.f.c n v z}
set instruction(MPY,CC)              {e}
set instruction(MPY,CCOMBO)          {e}
set instruction(MPY,SUFFIX)          {e}
set instruction(MPY,CYCLES)          2
```

FIG. 4




```
#setting up state
set instruction(MPY,AS,RXb)     {{maxint}                    {minint}                    }
set instruction(MPY,AS,RYb)     {{maxint}                    {minint}                    }
set instruction(MPY,AS,Cb)      {{0}                         {0}                         }
set instruction(MPY,AS,Vb)      {{0}                         {0}                         }
set instruction(MPY,AS,Nb)      {{0}                         {0}                         }
set instruction(MPY,AS,Zb)      {{0}                         {0}                         }

#specifying desired state.
set instruction(MPY,AS,RTa)     {{mpexpr[maxint]*[maxint]}   {mpexpr[minint]*[minint]}   }
set instruction(MPY,AS,Ca)      {{0}                         {0}                         }
set instruction(MPY,AS,Va)      {{0}                         {0}                         }
set instruction(MPY,AS,Na)      {{signOunsil}                {0}                         }
set instruction(MPY,AS,Za)      {{0}                         {signOunsil}                }
```